NONLINEAR OPTIMIZATION

Qingsha Cheng 程庆沙

Derivative-Free Optimization: Simplex Method

Remarks on derivative free optimization

Downhill simplex method: concept and algorithm

Derivative-Free Optimization

The **methods discussed so far require** first- and, sometimes, higher-order **derivative** information of the objective function

In many cases, **derivative** information is **not available** or **too expensive to obtain**; also, the objective function itself may be **non-differentiable** or even **discontinuous**

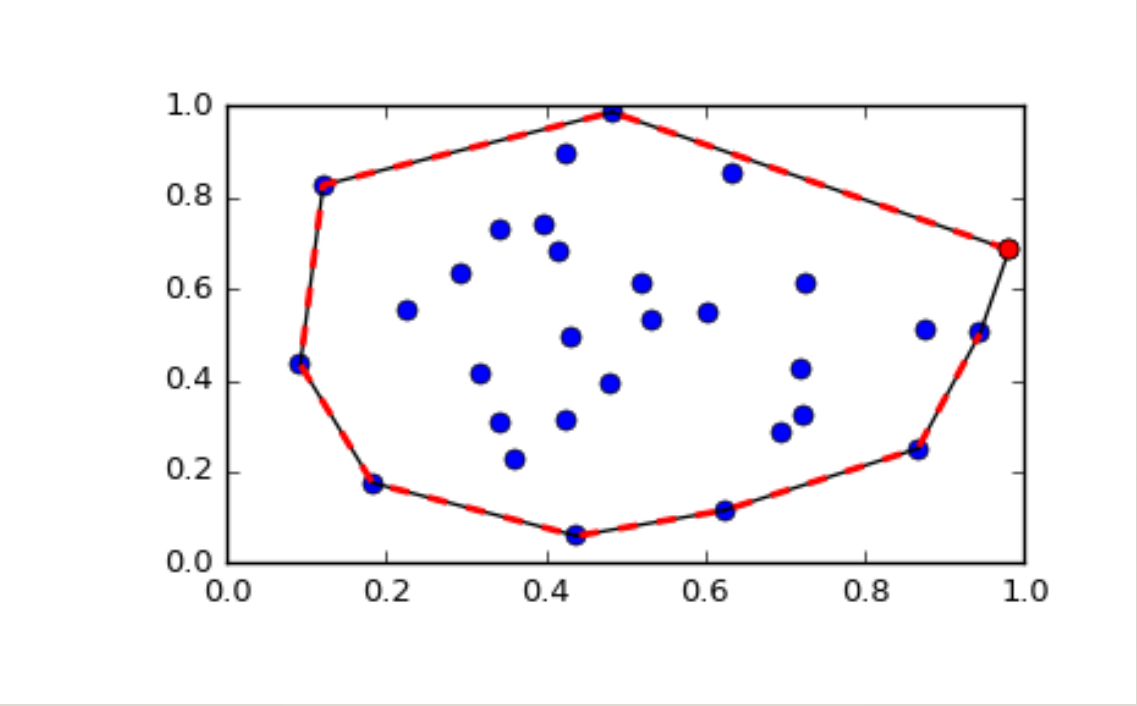
There is a number of methods that **do not require derivative** information, and we will deal with some of them while talking about **stochastic optimization techniques**

Here, we discuss a so-called **Nelder-Mead method**, also called a **downhill simplex method**, which is commonly used for **nonlinear optimization of smooth functions**

Simplex Method: Definitions

Definition: Let $X = \{x^{(i)} \in R^n, i = 1, 2, ..., k\}$. A *convex hull* $H(X)$ of X is defined as a set of all convex combinations of points from X, i.e.,

$$H(X) = \left\{ \sum_{i=1}^{k} a_i x^{(i)} : x^{(i)} \in X, a_i \in R, a_i \geq 0, \sum_{i=1}^{k} a_i = 1 \right\}$$



Simplex Method: Definitions

Definition: A set of points $x^{(i)} \hat{I} \; R^n$, $i = 1, 2, ..., k$, is *affinely independent* if the vectors $v^{(j)} = x^{(j+1)} - x^{(1)}$, $j = 1, ..., k\text{-}1$, are linearly independent.

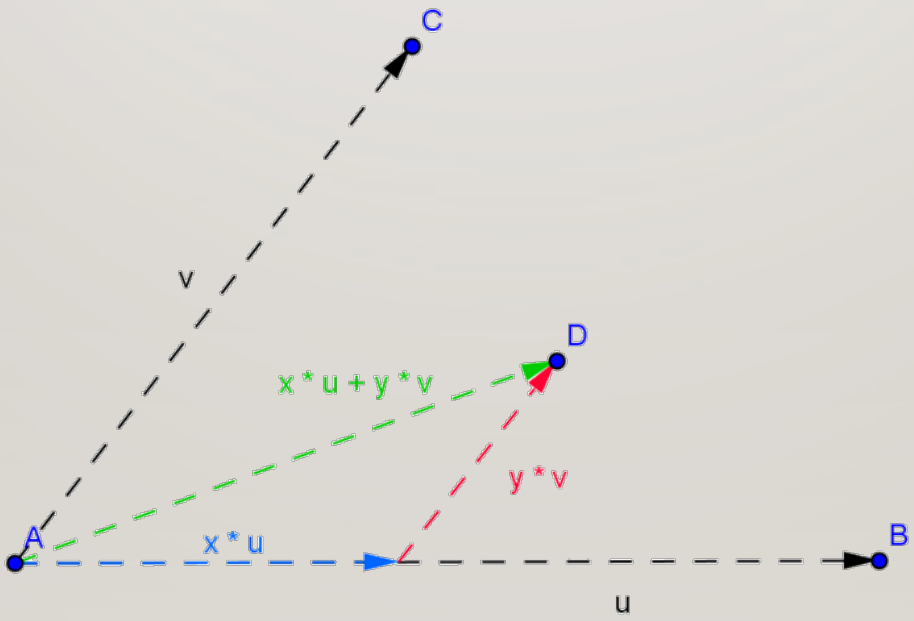


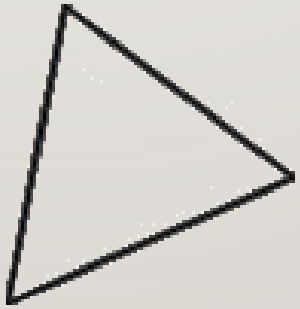
Figure 2: Point D is an affine combination of the points A, B, & C.

Simplex Method: Definitions

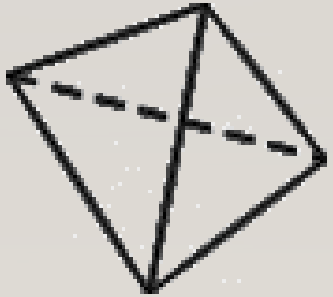
Definition: A *simplex* or *n-simplex* is a convex hull of a set of $n+1$ affinely independent points in R^m, $m \geq n$.

Examples:

triangle is a simplex in R^2



tetrahedron is a simplex in R^3



Downhill Simplex Method Algorithm

At any given iteration, the simplex method processes a set of simplex vertices and updates it based on the corresponding objective function values.

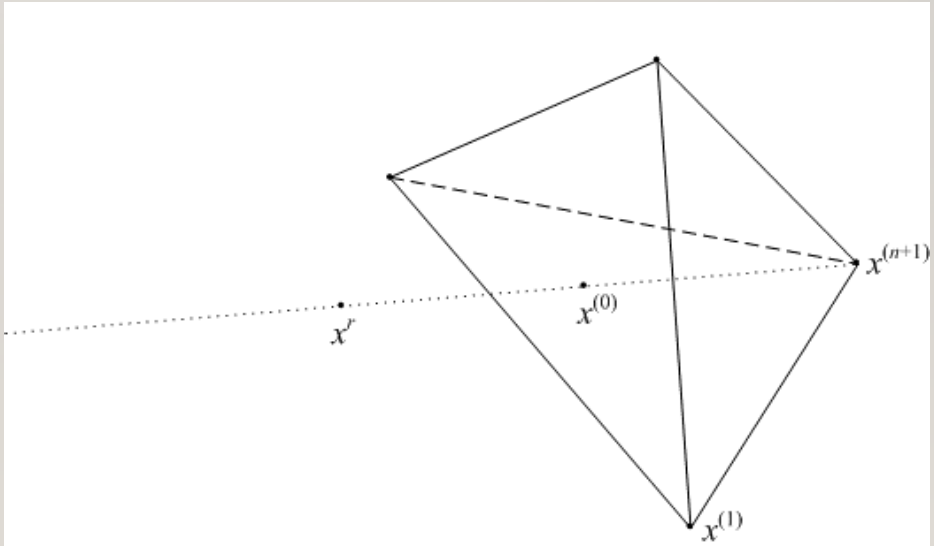
Steps:

1.order vertices so that $f(x^{(1)}) \leq f(x^{(2)}) \leq \ldots f(x^{(n+1)})$;

2.let $x^{(0)}$ be a center of gravity ($x^{(0)} = [x^{(1)} \ldots + x^{(n)}]/n$) of all points but $x^{(n+1)}$;

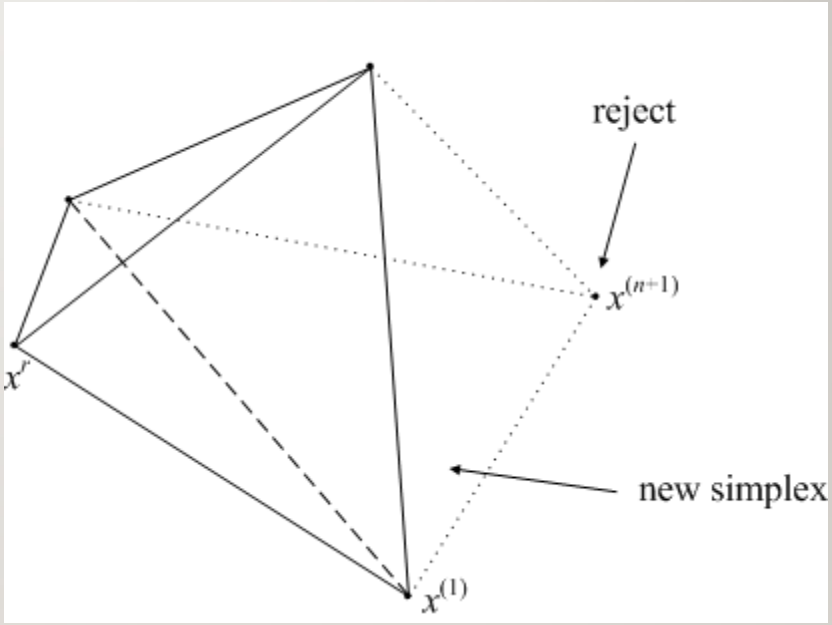
3.compute a *reflection* $x^r = (1+\alpha)x^{(0)} - \alpha x^{(n+1)}$

(typical value of α is 1)

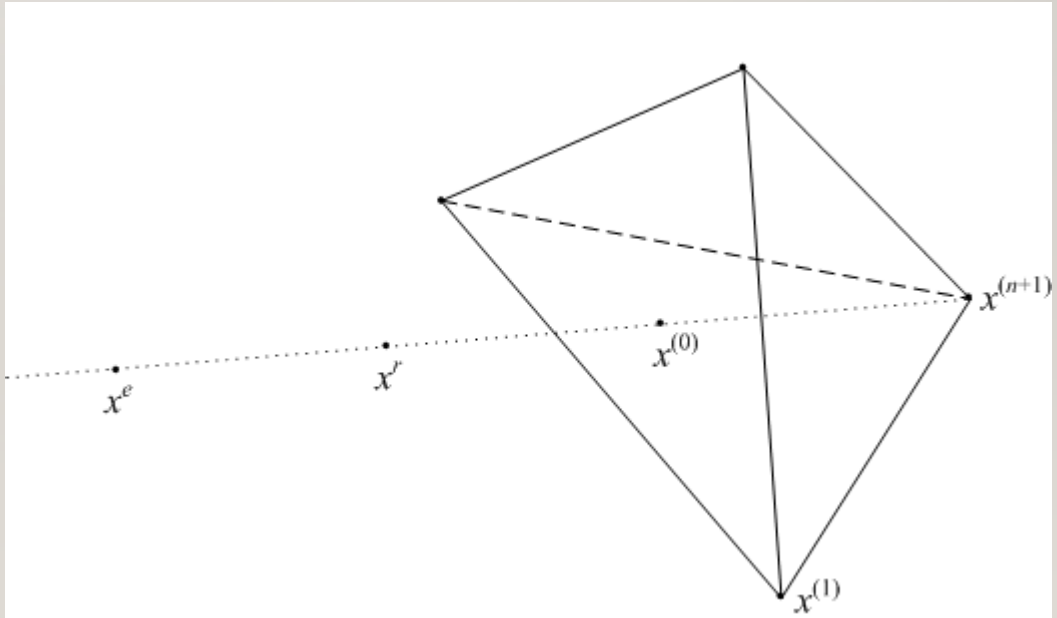


Downhill Simplex Method Algorithm

If $f(x^{(1)}) \leq f(x^r) < f(x^{(n)})$ reject $x^{(n+1)}$ and update the simplex using x^r

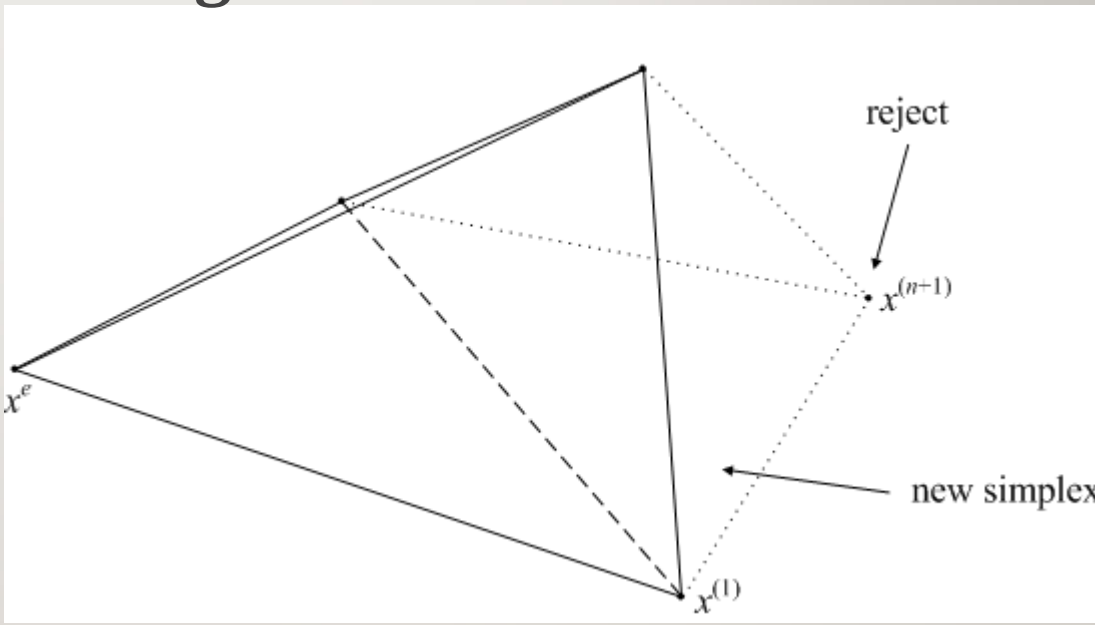


If $f(x^r) < f(x^{(1)})$ compute an *expansion* $x^e = \gamma x^r + (1-\gamma)x^{(0)}$ (typical value of γ is 2)

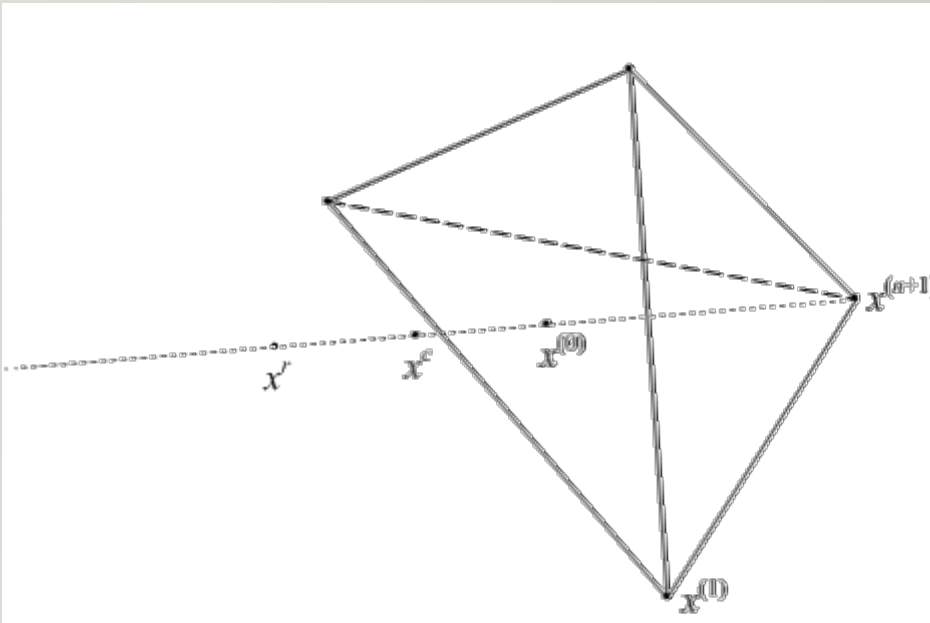


Downhill Simplex Method Algorithm

If $f(x^e) < f(x^r)$ reject $x^{(n+1)}$ and update the simplex using x^e

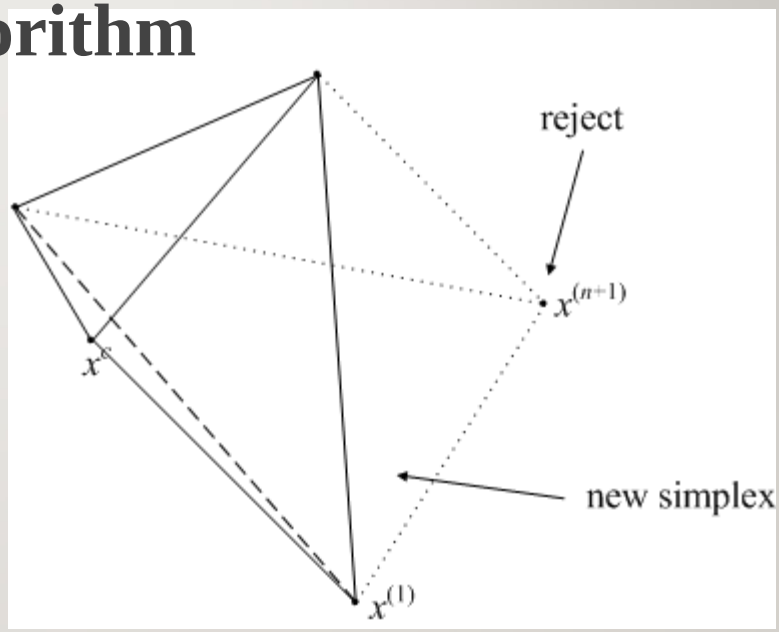


If $f(x^{(n)}) \leq f(x^r) < f(x^{(n+1)})$ compute a *contraction* $x^c = (1+\gamma)x^{(0)} - \gamma x^{(n+1)}$ (typical value of γ is 0.5)

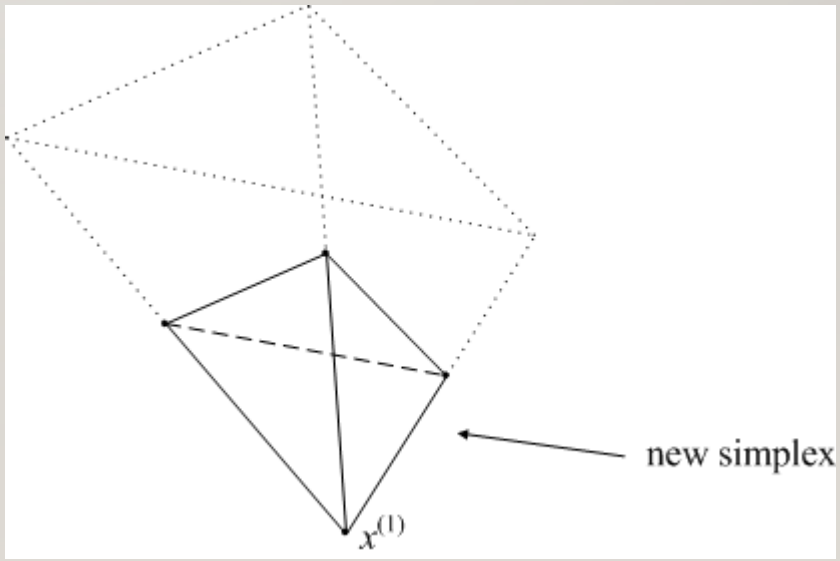


Downhill Simplex Method Algorithm

If $f(x^c) \leq f(x^r)$ reject $x^{(n+1)}$
and update the simplex
using x^c



If $f(x^c) > f(x^r)$ or $f(x^r) \geq f(x^{(n+1)})$
shrink the simplex:
$x^{(i)} = x^{(1)} + \sigma(x^{(i)} - x^{(1)})$, $i = 1, ..., n+1$
(typical value of σ is 0.5)

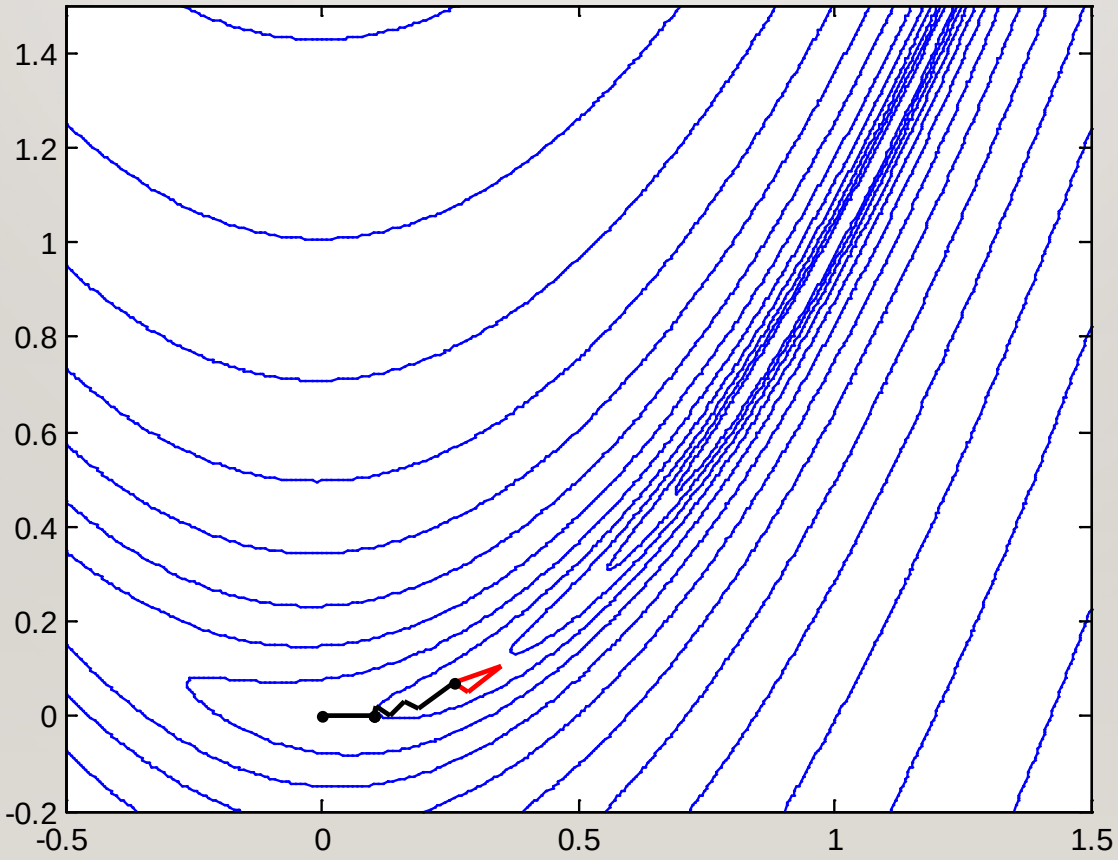


Downhill Simplex Method Algorithm

$S = \{x^{(1)}, ..., x^{(n)}\}$; $\alpha = 1$, $r = 2$, $\gamma = 0.5$, $\sigma = 0.5$;
while ~*termination_condition*
  order simplex vertices: $f(x^{(1)}) \le f(x^{(2)}) \le ... f(x^{(n+1)})$;
  calculate center of gravity $x^{(0)}$ of all points but $x^{(n+1)}$;
  $x^r = (1+\alpha)x^{(0)} - \alpha x^{(n+1)}$; % reflection
  if $f(x^{(1)}) \le f(x^r) < f(x^{(n)})$
    update simplex with x^r instead of $x^{(n+1)}$;
  elseif $f(x^r) < f(x^{(1)})$ % expansion
    $x^e = r x^r + (1-r)x^{(0)}$;
    if $f(x^e) < f(x^r)$
      update simplex with x^e instead of $x^{(n+1)}$;
    else
      update simplex with x^r instead of $x^{(n+1)}$;
    end
  elseif $f(x^{(n)}) \le f(x^r) < f(x^{(n+1)})$ % contraction
    $x^c = (1+\gamma)x^{(0)} - \gamma x^{(n+1)}$;
    if $f(x^c) \le f(x^r)$
      update simplex with x^c instead of $x^{(n+1)}$;
    else % shrink step
      $x^{(i)} = x^{(1)} + \sigma(x^{(i)} - x^{(1)})$ for $i = 1, 2, ..., n+1$;
    end
  else % shrink step
    $x^{(i)} = x^{(1)} + \sigma(x^{(i)} - x^{(1)})$ for $i = 1, 2, ..., n+1$;
  end
end

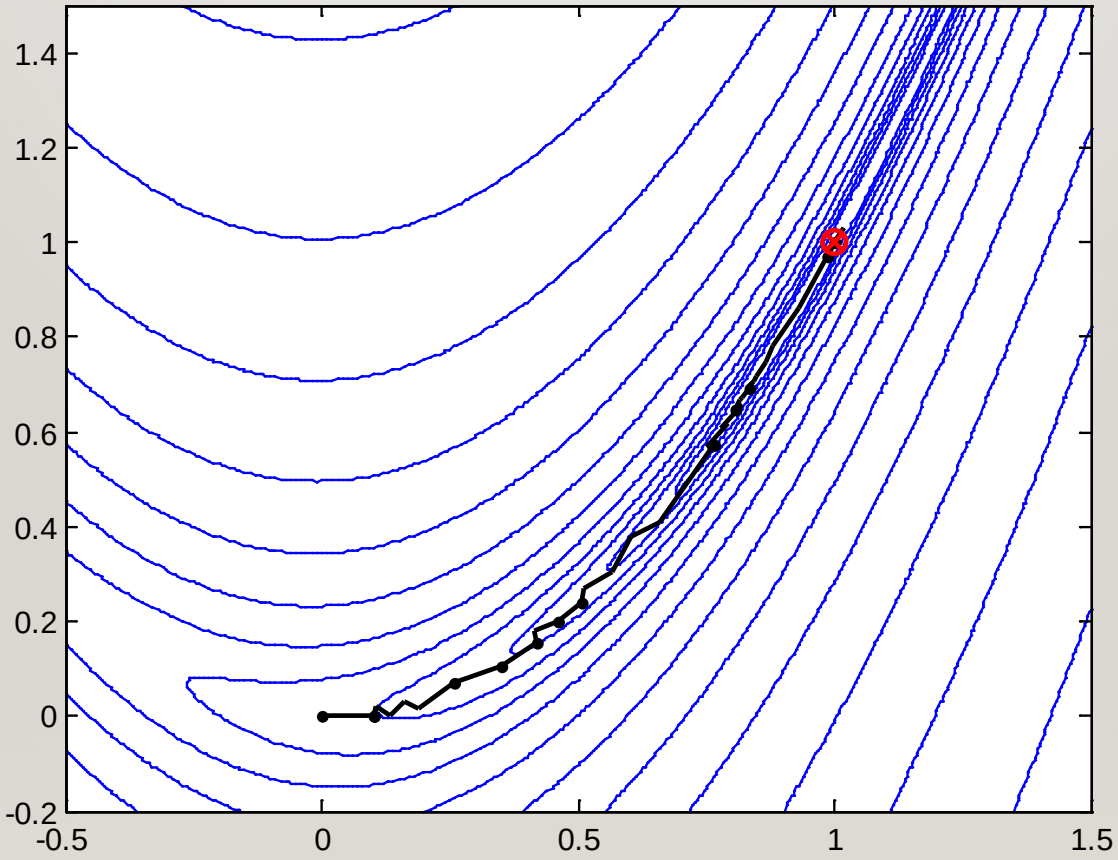
Downhill Simplex Method Algorithm

Example: downhill simplex algorithm minimizing the Rosenbrock function – in the middle of the algorithm run



Downhill Simplex Method Algorithm

Example: downhill simplex algorithm minimizing the Rosenbrock function – final result



Downhill Simplex Method Algorithm

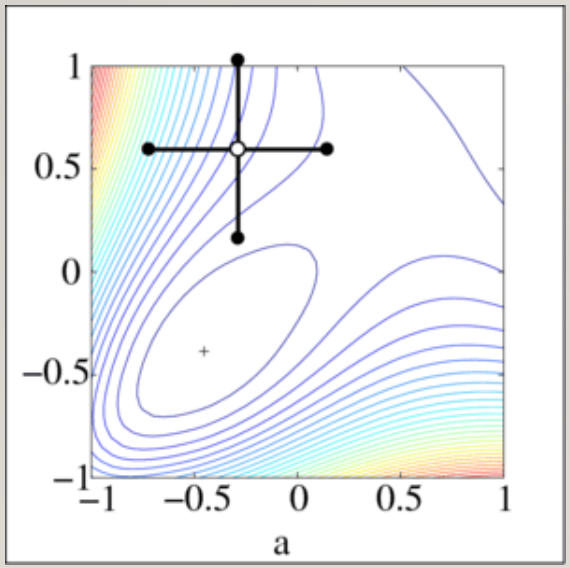
A video

Exercise 1: Pattern Search Algorithm

Develop and implement a simple pattern search algorithm that seeks for a minimum of a function *f* on a rectangular grid of variable size. Visualize the operation of the algorithm for the case when *f* is a function of two variables.

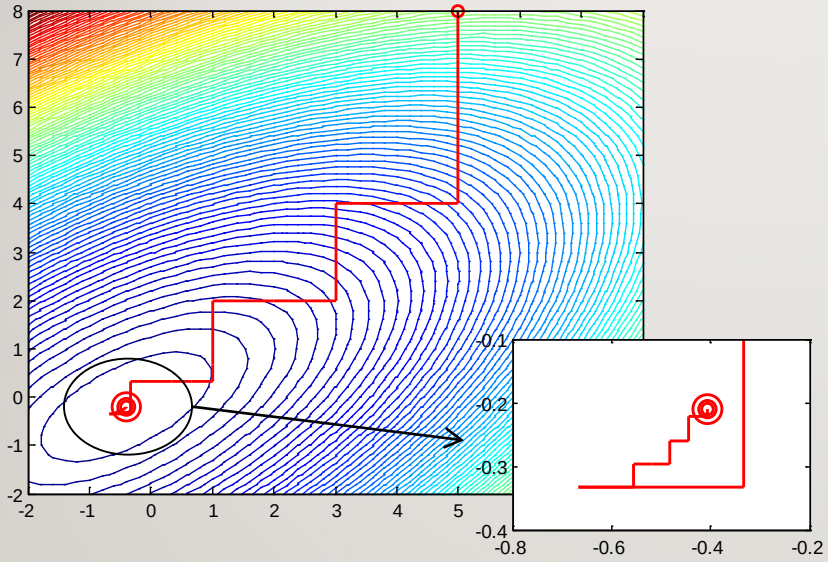
Test your function using the following cases:

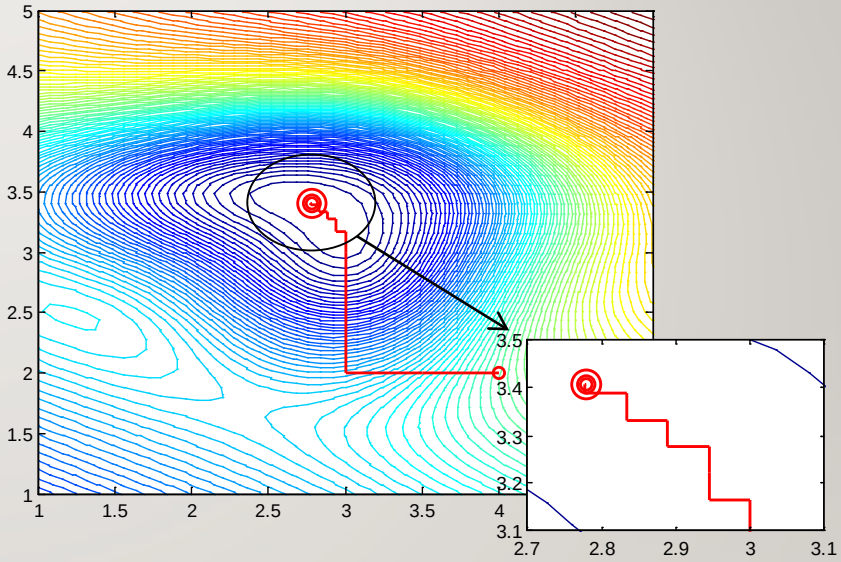
1. $f(x,y) = 2x^2 + 3y^2 - 3xy + x$, $-2 \le x, y \le 8$, starting point $[5\ 8]^T$
2. $f(x,y) = (1 - x)^2 + 5(x - y^2)^2$, $-0.5 \le x, y \le 1.5$, starting point $[0\ 0]^T$
3. $f(x,y) = (x + 2y)\cdot(1 - 0.9\cdot\exp(-0.3\cdot(x - 2.5)^2 - 2\cdot(y - 3.5)^2))\cdot(1 - 0.9\cdot\exp(-(x - 3)^2 - (y - 3)^2))$, $1 \le x, y \le 5$, starting point $[4\ 2]^T$
4. $f(x,y) = \exp(x/5) + \exp(y/3)$, $-10 \le x, y \le 10$, starting point $[5\ 8]^T$



They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small.

Simple pattern search algorithm: case 1 (left) and case 3 (right)





More involved pattern search algorithm: case 1 (left) and case 3 (right)